EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on April 27, 2009 has been entered.

3. Authorization for this examiner's amendment was given in a telephone interview with Jose R. de la Rosa (Reg. No. 34,810) on May 28, 2009.

4. Amend the Abstract line 2 as follows: replace "(OKD)" with --(ODK)--.

5. Amend the claims as follows:

1. (Currently Amended) A method of providing an open development kit (ODK) subsystem, ~~the open development kit having a virtual CPU adapted to execute a programmable logic controller (PLC) application program in a real-time operating environment and adapted to execute one or more extensions, the one or more extensions providing access into the scan cycle of the PLC and replaceable functionality to the operation of the PLC,~~ the method comprising the steps of:

executing a programmable logic controller (PLC) application program in the ODK subsystem, wherein the ODK subsystem having a virtual CPU, the virtual CPU executing the PLC program in a real-time operating environment and executing one or more extensions, the one or more extensions providing access into a scan cycle of the PLC program and replaceable functionality to the operation of the PLC program;

~~checking whether there are requests for application execution in the PLC;~~

sending a specific formatted load application request message from the PLC program to ~~an open development kit (ODK)~~ the ODK subsystem;

converting at the ODK subsystem the specific formatted load application request message to a generic formatted load ~~application~~ request message;

sending the generic formatted load request message from the ODK subsystem to an application for requesting the one or more extensions ~~an extension~~ to be loaded by the application; and

requesting execution of the application, including checking whether there are requests for application execution in the PLC program and, if so, sending a specific formatted execution request from the PLC program to the ODK subsystem, converting the specific formatted execution request to a generic execution request, and sending the generic execution request from the ODK subsystem to the application, the application comprising an Application Wizard, the Application Wizard ~~adapted to provide~~ providing a first mode and a second mode[[,]];

checking whether any requests for information are waiting in the application and, if so, requesting information from the PLC program by the application;

executing a function in the ODK subsystem specified by the application;

performing a task in the PLC program associated with the executed function in the ODK subsystem;

returning a specific formatted response from the PLC program to the ODK subsystem, the ODK subsystem converting the specific formatted response to a generic response; and

returning the generic response from the ODK subsystem to the application, the ODK subsystem comprising an ODK Real-Time (RTX) Proxy Dynamic Link Library (DLL) ~~RTX Proxy DLL~~ usable in the deterministic environment, the ODK RTX Proxy DLL ~~adapted to communicated~~ communicating through shared memory with an ODK System Block (SB) Add-on DLL ~~SB Add-ion DLL~~ in the non-deterministic environment, wherein, in the first mode, the Application Wizard generates a Real Time Dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and

the deterministic environment is a WinAC RTX environment.

2. (Currently Amended) The method of claim 1, further comprising the steps of:

initializing callback function pointers for use by the application as callback functions into the ODK subsystem; and

initializing the one or more extensions ~~extension~~ after the one or more extensions ~~extension~~ is loaded.

3. (Currently Amended) The method of claim 1, wherein the one or more extensions ~~extension~~ is a dynamic ~~load~~ link library (DLL).

4. (Original) The method of claim 1, further comprising the step of checking whether a stop to run transition has occurred in the PLC program and if so, sending a specific formatted activate application message from the PLC program to the ODK subsystem.

5. (Previously Presented) The method of claim 4, further comprising the step of calling an activate function in the application by the ODK subsystem thereby permitting scan cycle execution, the Real Time Dynamic Link Library project comprising a helper class ~~adapted to~~ ~~access~~ accessing data formats of Step 7 type data from a buffer that is passed between a central processing unit and the application.

6. (Cancelled)

7. (Currently Amended) The method of ~~claim 6~~ claim 1, further comprising the step of executing the generic execution request by the application.

8. (Currently Amended) The method of ~~claim 6~~ claim 1, further comprising the steps of:

sending a generic response from the application to the ODK subsystem;

converting the generic response to a specific formatted response; and

sending the specific formatted response from the ODK subsystem to the PLC program.

9. (Original) The method of claim 8, further comprising the step of returning at least one of data and a control block from the application to the ODK subsystem, and from the ODK subsystem to the PLC program.

10. (Cancelled)

11. (Currently Amended) The method of ~~claim 10~~ claim 1, wherein in the requesting information ~~step~~ from the PLC program the application uses a call back pointer to generically request information and the executing the function step executes the function in the ODK subsystem corresponding to the callback pointer.

12. (Currently Amended) The method of claim 11, in the executing ~~a~~ the function step, the function is provided by a dynamic link library.

13. (Cancelled)

14. (Currently Amended) The method of ~~claim 10~~ claim 1, wherein when the checking whether any requests for information are waiting in the application determines that there are no requests for information waiting, ~~and~~ the method further comprising:

waiting until the PLC program transitions from a run state to a stop state;

sending a deactivate request from the PLC program to the ODK subsystem; and

calling a deactivate function in the application.

15. (Currently Amended) The method of claim 14, wherein when a memory clear or PLC program shutdown occurs, the step of calling a release function in the application and unloading the one or more extensions ~~extension~~ occurs.

16. (Currently Amended) The method of claim 1, wherein ~~in~~ the sending ~~a~~ the load application request message from the PLC program is sent from at least one of a soft PLC, a slot PLC and a hard PLC.

17. (Currently Amended) A system for open development that comprises a computer-readable storage medium encoded with a computer program, the system further ~~computer program~~ comprising:

a processor;

one or more extensions ~~adapted to provide~~ providing real-time services in a real-time operating environment; and

a virtual CPU ~~adapted to execute~~ executing a programmable logic controller (PLC) application program in the real-time operating environment and ~~adapted to~~ further executing the steps of:

receiving a specific formatted load application request message from the PLC program, convert the specific formatted load application request message to a generic formatted load request message, send the generic formatted load request message to an application for requesting the one or more extensions to be loaded by the application; and

~~check~~ checking whether there are requests for application execution in the PLC program and, if so, receive a specific formatted execution request from the PLC program, convert the specific formatted execution request to a generic execution request, send the generic execution request to ~~the~~ an application, and execute the one or more extensions, ~~the system comprising~~ wherein the application comprising an Application Wizard, the Application Wizard ~~adapted to provide~~ providing a first mode and a second mode, and

~~check~~ checking whether any requests for information are waiting in the application and, if so, request information from the PLC program, execute a function specified by the application, initiate performance of a task in the PLC program associated with the executed function, receive a specific formatted response from the PLC program, convert the specific formatted response to a generic response, and return the generic response to the application using a Real Time (RTX) Proxy Dynamic Link Library (DLL) ~~RTX Proxy DLL~~, the RTX Proxy DLL usable in the deterministic environment and ~~adapted to communicated~~ communicating through shared memory with a System Block (SB) Add-on DLL in the non-deterministic environment,

wherein, in the first mode, the Application Wizard generates a Real Time Dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and wherein the one or more extensions provide access into the scan cycle of the PLC program and to provide replaceable functionality to the operation of the PLC program, and

the deterministic environment is a WinAC RTX environment.

18. (Currently Amended) The system of claim 17, further comprising:

a system block loader ~~adapted to load~~ loading system blocks, the system blocks including at least one of a system function block, a system function, a system data block; and

an Open Development Kit (ODK) ~~ODK~~ SB Add-on dynamic link library (DLL) for implementing a common object module (COM) interface for the virtual CPU and system block loader.

19. (Original) The system of claim 17, wherein the one or more extensions are dynamic link libraries.

20. (Currently Amended) A system ~~that comprises~~ comprising a computer-readable storage medium encoded with a computer program, the computer program ~~adapted to provide~~ providing an open development kit (ODK) subsystem, the system ~~computer program~~ further comprising:

~~a~~ means for ~~executing a programmable logic controller (PLC) application program in the real-time operating environment and adapted to execute the one or more extensions in virtual CPU~~ executing a programmable logic controller (PLC) application program in the ODK subsystem, the ODK subsystem having a virtual CPU executing the PLC program in a real-time operating environment and executing one or more extensions, the one or more extensions providing access into the scan cycle of the PLC program and replaceable functionality to the operation of the PLC program;

~~a means for determining whether there are requests for application execution in the PLC;~~

~~a~~ means for sending a specific format load application request message from ~~a programmable logic controller (PLC)~~ the PLC program to ~~an open development kit (ODK) subsystem,~~ the ODK subsystem ~~in response to determining requests for application execution are in the PLC~~;

~~a~~ means for converting the specific format load application request to a generic format load request message by the ODK subsystem;

~~a~~ means for sending the generic format load request message from the ODK subsystem to an application requesting the one or more extensions ~~an extension~~ to be loaded by the application;

~~a~~ means for requesting execution of the application, including a means for checking whether there are requests for application execution in the PLC program and, if so, a means for sending a specific formatted execution request from the PLC program to the ODK subsystem, a means for converting the specific formatted execution request to a generic execution request, and a means for sending the generic execution request from the ODK subsystem to the application, the application comprising an Application Wizard, the Application Wizard ~~adapted to provide~~ providing a first mode and a second mode;

~~a~~ means for determining whether any requests for information are waiting in the application and, if so, requesting information from the PLC program by the application;

~~a~~ means for executing a function in the ODK subsystem specified by the application;

~~a~~ means for performing a task in the PLC program associated with the executed function in the ODK subsystem;

~~a~~ means for returning a specific formatted response from the PLC program to the ODK subsystem, the ODK subsystem converting the specific formatted response to a generic response; and

~~a~~ means for returning the generic response from the ODK subsystem to the application, the ODK subsystem comprising an ODK Real-Time (RTX) Proxy Dynamic

Link Library (DLL) ~~RTX Proxy DLL~~ usable in the deterministic environment, the ODK RTX Proxy DLL ~~adapted to communicated~~ communicating through shared memory with an ODK System Block (SB) Add-on DLL ~~SB Add-ion DLL~~ in the non-deterministic environment, wherein, in the first mode, the Application Wizard generates a Real Time dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and

the deterministic environment is a WinAC RTX environment.[[.]]

21. (Currently Amended) The system of claim 20, further comprising ~~the steps of~~:

~~a~~ means for initializing callback function pointers for use by the application as callback functions into the ODK subsystem; and

~~a~~ means for initializing the one or more extensions ~~extension~~ after the one or more extensions ~~extension~~ is loaded.

22. (Original) The system of claim 20, further comprising ~~a~~ means for checking whether a stop to run transition has occurred in the PLC program and if so, ~~a~~ means for sending a specific activate application message from the PLC program to the ODK subsystem.

23. (Original) The system of claim 22, further comprising ~~a~~ means for calling an activate function in the application by the ODK subsystem thereby permitting scan cycle

execution.

24. (Cancelled)

25. (Currently Amended) The system of ~~claim 24~~ claim 20, further comprising ~~a~~ means for executing the generic formatted execution request by the application.

26. (Original) The system of claim 25, further comprising:

~~a~~ means for sending a generic response from the application to the ODK subsystem;

~~a~~ means for converting the generic response to a specific formatted response; and

~~a~~ means for sending the specific formatted response from the ODK subsystem to the PLC program.

27. (Original) The system of claim 26, further including ~~a~~ means for returning at least one of data and a control block from the application to the ODK subsystem, and from the ODK subsystem to the PLC program.

28. (Cancelled)

29. (Currently Amended) The system of ~~claim 28~~ claim 20, wherein the means for requesting information by the application uses a call back pointer to request information and the means for executing the function in the ODK subsystem executes the function in the ODK subsystem corresponding to the callback pointer.

30. (Original) The system of claim 29, wherein the function is provided by a dynamic link library (DLL).

31. (Original) The system of claim 20, wherein the PLC program is at least one of a soft PLC, a hard PLC, and a slot PLC.

32. (Cancelled).

33. (Currently Amended) A computer program product comprising a computer storage medium having readable program code embodied in the medium, the computer program product further comprising a kit for open development (ODK) subsystem, the ODK subsystem comprising: ~~A computer readable~~ storage ~~medium encoded with a computer program, the computer program comprising a kit for open development (ODK) subsystem, the kit for open development ODK subsystem comprising:~~

~~a~~ means for executing a programmable logic controller (PLC) application program in a real-time operating environment ~~adapted to execute~~ ~~the~~ executing one or more extensions in a virtual CPU, the one or more extensions providing access into the scan cycle of the PLC program and replaceable functionality to the operation of the PLC program;

~~a~~ means for receiving a specific formatted load application request message from the PLC program;

~~a~~ means for converting the specific formatted load application request message to a generic formatted load request message;

~~a~~ means for sending the generic formatted load request message to an application for requesting the one or more extensions to be loaded by the application;

~~a~~ means for determining whether there are requests for application execution in the PLC program,

~~a~~ means for receiving a specific formatted message from ~~a programmable logic controller (PLC),~~ the PLC program in response to determining requests for application execution are in the PLC program, wherein the specific formatted message is a request for application execution;

~~a~~ means for converting the specific formatted message to a generic formatted message;

~~a~~ means for sending the generic formatted message to an application for execution, the application comprising an Application Wizard, the Application Wizard ~~adapted to provide~~ providing a first mode and a second mode;

~~a~~ means for checking whether any requests for information are waiting in the application;

~~a~~ means for requesting information from the PLC program by the application in response to determine requests for information are waiting in the application;

~~a~~ means for executing a function in the ODK subsystem specified by the application;

~~a~~ means for performing a task in the PLC program associated with the executed function in the ODK subsystem;

~~a~~ means for returning a specific formatted response from the PLC program to the ODK subsystem, the ODK subsystem converting the specific formatted response to a generic response; and

~~a~~ means for returning the generic response from the ODK subsystem to the application, the ODK subsystem comprising an ODK Real-Time (RTX) Proxy Dynamic Link Library (DLL) ~~RTX Proxy DLL~~ usable in the deterministic environment, the ODK RTX Proxy DLL ~~adapted to communicated~~ communicating through shared memory with an ODK System Block (SB) Add-on DLL ~~SB Add-ion DLL~~ in the non-deterministic environment, wherein, in the first mode, the Application Wizard generates a Real Time dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and

the deterministic environment is a ~~WinRTX~~ WinAC RTX environment.

34. (Previously Presented) The ~~kit~~ computer program product of claim 33, wherein the ODK subsystem further comprising:

~~a~~ means for receiving the generic formatted message from the application;

~~a~~ means for converting the generic formatted message to the specific formatted message; and

~~a~~ means for sending the specific formatted message to the PLC program.

35. (Original) The ~~kit~~ computer program product of claim 34, wherein at least one of the means includes using a dynamic link library that loads replaceable functionality.

36. (Original) The ~~kit~~ computer program product of claim 34, wherein the generic formatted message is one of a response message and a request for information message.

37. (Original) The ~~kit~~ computer program product of claim 34, wherein the specific formatted message is sent to at least one of a soft PLC, a hard PLC and a slot PLC.

38. (Currently Amended) A computer program product comprising a computer storage medium having readable program code embodied in the medium, the computer program product ~~includes~~ comprising:

~~a first computer program code to check whether there are requests for application execution in the PLC;~~

a ~~second~~ first computer program code to send a specific formatted load application request message from a programmable logic controller (PLC) to an open development kit (ODK) subsystem;

a ~~third~~ second computer program code to convert the specific formatted load application request message to a generic formatted load application request message;

a ~~fourth~~ third computer program code to send the generic formatted load ~~application~~ request message from the ODK subsystem to an application for requesting an extension to be loaded by the application; and

a ~~fifth~~ fourth computer program code to request execution of the application, check whether there are requests for application execution in the PLC and, if so, send a specific formatted execution request from the PLC to the ODK subsystem, convert the specific formatted execution request to a generic execution request, and send the generic execution request from the ODK subsystem to the application, the application comprising an Application Wizard, the Application Wizard ~~adapted to provide~~ providing a first mode and a second mode;

a ~~sixth~~ fifth computer program code to check whether any requests for information are waiting in the application and, if so, request information from the PLC by the application, execute a function in the ODK subsystem specified by the application, and perform a task in the PLC associated with the executed function in the ODK subsystem;

a ~~seventh~~ sixth computer program code to return a specific formatted response from the PLC to the ODK subsystem, the ODK subsystem converting the specific formatted response to a generic response, and return the generic response from the ODK subsystem to the application, the ODK subsystem comprising an ODK Real-Time RTX Proxy Dynamic Link Library DLL ~~RTX Proxy DLL~~ usable in the deterministic environment, the ODK RTX Proxy DLL ~~adapted to~~ ~~communicated~~ communicating through shared memory with an ODK System Block (SB) Add-on DLL ~~SB Add-ion DLL~~ in the non-deterministic environment; and

an ~~eight~~ seventh computer program code to provide a virtual CPU ~~adapted to execute~~ executing a programmable logic controller (PLC) application program in the real-time operating environment and ~~adapted to execute~~ executing the one or more extensions, the one or more extensions providing access into the scan cycle of the PLC and replaceable functionality to the operation of the PLC;

wherein, in the first mode, the Application Wizard generates a Real Time dynamic Link Library project that is usable in a deterministic environment with fixed scan cycles, and in the second mode, the Application Wizard generates a Dynamic Link Library project that is usable in a non-deterministic environment with non-fixed scan cycles, and

the deterministic environment is a WinAC RTX environment.

6. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled “Comments on Statement of Reasons for Allowance.”

CONCLUSION

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent 5,038,318 to Roseman, US 6,243,857 to Logan, III et al., U.S. Patent 6,564,242, to Bonet et al., U.S. Patent 6,618,817 B1 to Armstrong, and U.S. Patent 7,478,406 B2 to Pauly et al., disclose a mechanism for communicating real-time data in between a programmable logic controller and an application program in a distributed computing environment.

U.S. Patent 6,816,865 B2 and U.S. Patent 7,373,349 B2 to O'Brien et al., disclose a mechanism for mapping multiple request/response messages to a format understandable to an appropriate application(s).

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to KimbleAnn Verdi whose telephone number is (571)270-1654. The examiner can normally be reached on Monday-Friday 7:30am-5:00pm EST.

9. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on (571) 272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

10. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Hyung S. Sough/
Supervisory Patent Examiner, Art Unit 2194
06/02/09
May 29, 2009

KV